Ahsanullah University of Science & Technology
Department of Computer Science & Engineering

Project Title

Computer Graphics Lab (CSE 4204)

Project Final Report

Submitted By:

Md. Tahiadur Rahman	20200204003
Adiba Amin	20200204012

Project Requirements:

The objective of this project is to create an immersive 3D simulation of an asteroid field in space. The project requires developing an interactive environment using WebGL through the Three.js library. It integrates several advanced graphics techniques such as custom GLSL shaders for realistic lighting effects, dynamic object generation (asteroids and meteors), interactive camera controls, and collision avoidance mechanisms. Additionally, it features a spaceship model with keyboard-controlled movement and interactive texture switching.

Software Platform:

Tools and Technologies Used:

- **HTML5 & CSS3:** For structuring and styling the web page.
- **JavaScript:** Core programming language for logic and interactivity.
- **Three.js (v0.143.0):** For rendering 3D graphics.
- **OrbitControls:** To allow user-friendly camera orbiting.
- **GLTFLoader:** For loading the 3D spaceship model.
- **WebGL:** The underlying technology for GPU-accelerated rendering.

Project Features:

This project demonstrates a range of advanced graphics and interactive features:

- **3D Scene Setup:** Creation of a dynamic 3D scene complete with a starry background, perspective camera, and a WebGL renderer that resizes with the browser window.
- **Custom Shaders for Asteroids:** Utilization of custom GLSL vertex and fragment shaders to provide realistic lighting and texturing for each asteroid.
- **Dynamic Asteroid Field:** Generation of 50 asteroids with 3 different textures, randomized positions, rotations, sizes, and persistent drift velocities to simulate a natural asteroid field.
- **Spaceship Model & Texture Switching:** Implementation of a GLTF-loaded spaceship that can be navigated using keyboard controls. Users can change the spaceship texture interactively via mouse clicks.
- **Interactive Controls:** Integration of both keyboard (for spaceship navigation) and mouse controls (using OrbitControls for camera movement) to enhance user interactivity.

- **Camera following the spaceship:** The camera follows the spaceship in all directions along with the spaceship's movement and with the mouse controls the camera can view the scene in all directions as well.

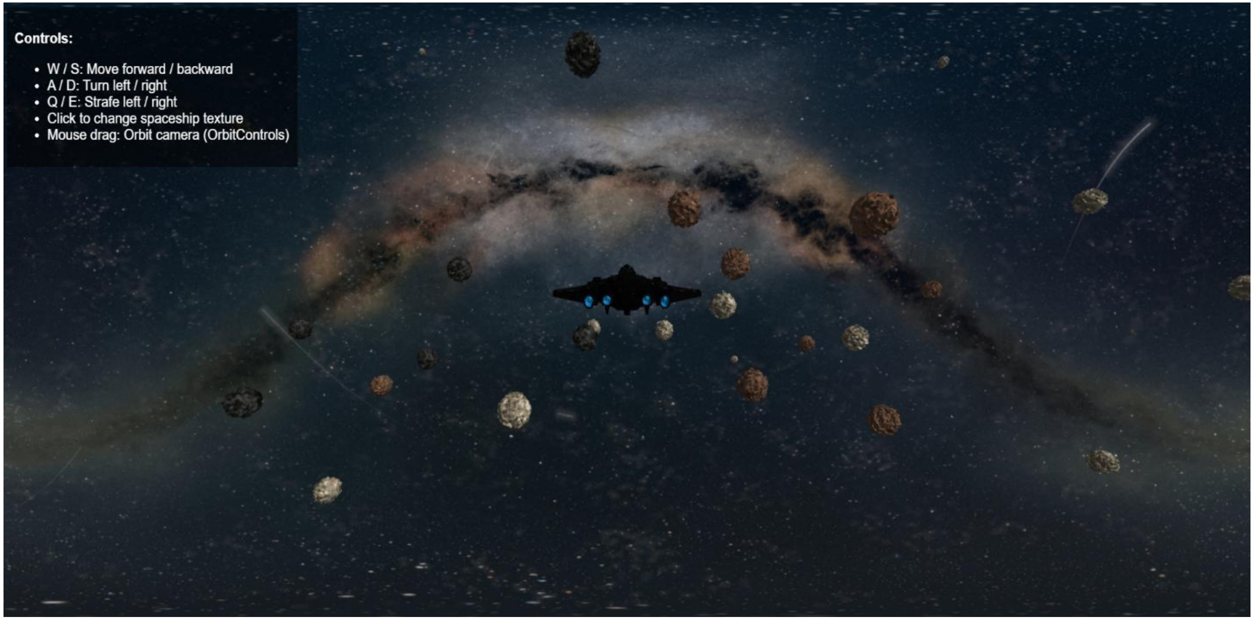
- **Meteor Shooting Logic:** Periodic generation of meteors with randomized movement patterns, adding dynamism to the scene.

- **Collision Avoidance Mechanisms:** Basic collision avoidance between the spaceship and other objects (asteroids and meteors) as well as among asteroids themselves.

#	Features	Status
1	3D Scene Setup	Implemented
2	Custom Shader for Asteroids	Implemented
3	Dynamic Asteroid Field Generation	Implemented
4	Spaceship Model & Texture Switching	Implemented
5	Interactive Controls (Keyboard & Orbit)	Implemented
6	Camera following the spaceship	Implemented
7	Meteor Shooting Logic (Additional Feature)	Implemented
8	Collision Avoidance Mechanisms (Additional Feature)	Implemented

Table 01: Project Feature Table

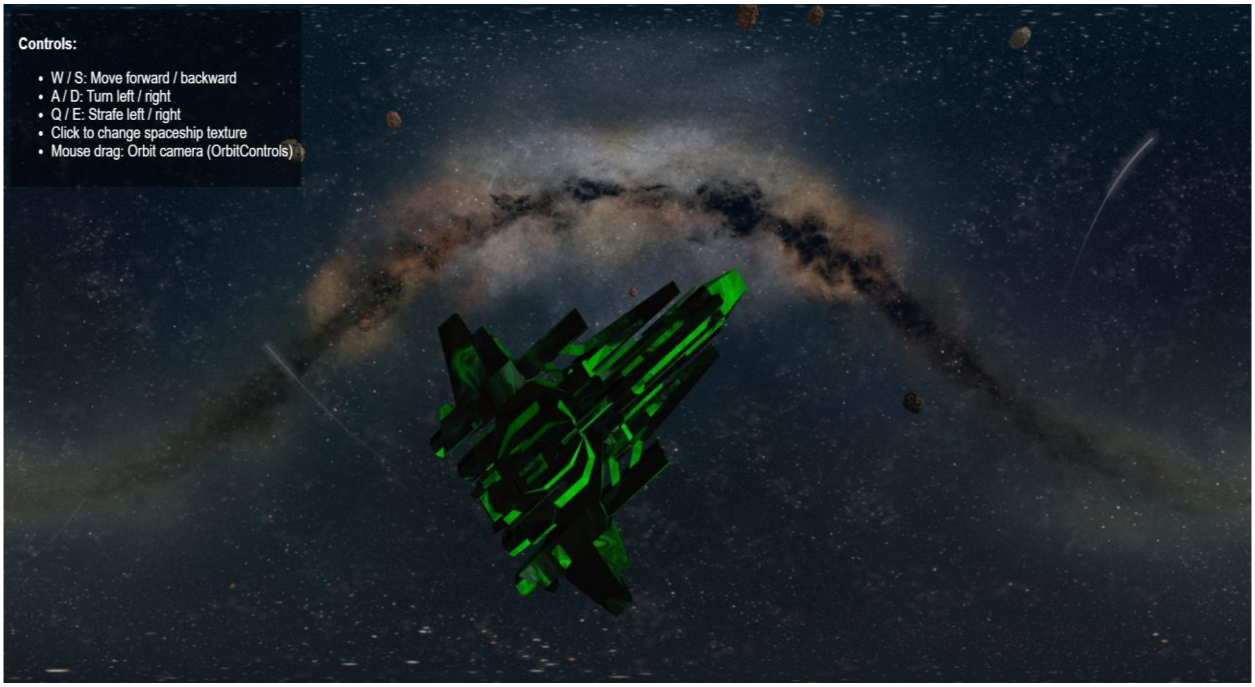
Snapshots:



Controls:

- W / S: Move forward / backward
- A / D: Turn left / right
- Q / E: Strafe left / right
- Click to change spaceship texture
- Mouse drag: Orbit camera (OrbitControls)

Figure 01: Initial Scene with a starry background, a spaceship and randomly generated, rotating and moving asteroids



Controls:

- W / S: Move forward / backward
- A / D: Turn left / right
- Q / E: Strafe left / right
- Click to change spaceship texture
- Mouse drag: Orbit camera (OrbitControls)

Figure 02: Change in Spaceship texture and camera angle

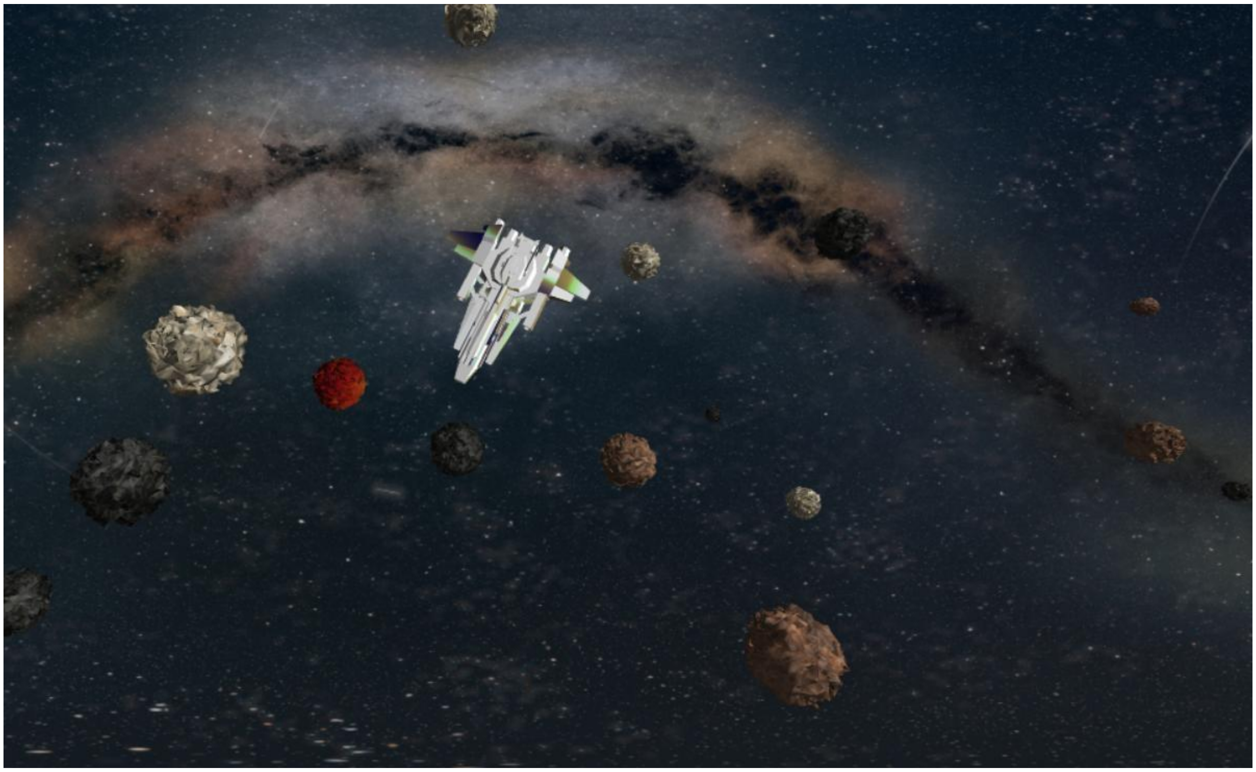


Figure 03: Randomly meteors falling at intervals, another change in spaceship texture and camera angle

Contribution:

The contribution in work by the team members is illustrated in the table below:

Feature	Md. Tahiadur Rahman (20200204003)	Adiba Amin (20200204012)
3D Scene and camera Setup	Structured the scene, set up WebGL renderer, and loaded background textures.	Integrated ambient and directional lighting, camera setup for the scene.
Custom Shaders for Asteroids	Developed GLSL vertex and fragment shaders and managed shader lighting parameters.	Configured texture mapping and tested shader performance with multiple textures.
Dynamic Asteroid Field & Meteor Generation	Implemented asteroid positioning, movement, and drift mechanics.	Designed meteor shooting logic, random generation, and movement patterns.
Interactive Controls	Designed and implemented keyboard controls for spaceship navigation.	Developed mouse interactions for camera orbiting and texture switching.
Collision Avoidance Mechanisms	Handled asteroid-asteroid collision avoidance mechanics.	Implemented asteroid and meteor collision logic with the spaceship.
Overall Integration & Testing	Debugged, optimized, and refined spaceship movement logic.	Debugged, optimized, and refined camera and interactive mechanics.

Future Work:

To further enhance the project, the following improvements are proposed:

- **Advanced Collision Detection:** Incorporate more sophisticated physics algorithms to handle collisions between objects realistically.
- **Enhanced Visual Effects:** Integrate particle systems and additional post-processing effects (e.g., bloom, depth of field) for increased visual fidelity.
- **Performance Optimization:** Optimize the rendering pipeline to handle larger numbers of objects and complex scenes without compromising performance.
- **Sound Integration:** Add background music and sound effects to provide a more immersive experience.
- **Virtual Reality (VR) Support:** Extend the project for VR environments to offer a fully immersive experience.
- **User Interface Enhancements:** Develop a more detailed UI to display real-time information about spaceship status, score, or other gameplay elements.